

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Розробка мобільних застосунків під Android

Лабораторна робота №6

Виконав:
студент групи Ю-23
Корбут М. Я.
Залікова книжка №2313
Перевірив
Орленко С. П.

Лабораторна робота №6

Опис програми

Для виконання лабораторної роботи №6 я вирішив розширити додаток за попередньої роботи з використанням геолокації (GPS). Базова версія додатку використовує сенсор обертання (Sensor.TYPE_ROTATION_VECTOR) для визначення кута нахилу автомобіля. Розширена версія отримує дані з GPS, обчислює швидкість у км/год, згладжує її та візуалізує значення як у вигляді тексту, так і у вигляді стрілки на шкалі спідометра. Також реалізовано кольорове кодування залежно від швидкості.

Основні компоненти додатка

- Java (Android SDK)
- Сенсори Android (SensorManager, RotationVectorSensor)
- GPS (LocationManager)
- Згладжування даних (експоненціальне фільтрування)
- Збереження налаштувань (SharedPreferences)
- Відображення графіки (ImageView + обертання стрілки).

GPS та локаційні дані в Android

У цьому додатку використовуються GPS та локаційні сервіси Android для отримання координат пристрою в режимі реального часу. Для роботи з геоданими необхідно вказати відповідні дозволи в AndroidManifest.xml:

- ACCESS_FINE_LOCATION — для точного місцезнаходження (GPS),
- ACCESS_COARSE_LOCATION — для менш точної інформації (через Wi-Fi, мобільні мережі).

Оскільки з Android 6.0 (API 23) введено модель дозволів часу виконання, додаток також перевіряє та запитує дозволи безпосередньо під час запуску.

Для отримання місцезнаходження використовується клас `FusedLocationProviderClient`, який забезпечує ефективний і енергозберігаючий спосіб доступу до геоданих. Отримане значення `Location` включає такі параметри, як широта, довгота, висота, точність, напрям руху (bearing) і швидкість.

Швидкість (`getSpeed()`) повертається в метрах за секунду (м/с) і розраховується на основі зміни координат та часу між запитами. Для точного визначення швидкості необхідне стабільне GPS-з'єднання та рух пристрою — у стані спокою швидкість може бути нульовою або неточною.

Ключові фрагменти коду

Обробка сенсора нахилу

```
rSensorManager.getRotationMatrixFromVector(rotationMatrix, event.values);
SensorManager.getOrientation(rotationMatrix, orientationAngles);
float pitchRad = orientationAngles[1];
float rawTilt = (float) Math.toDegrees(pitchRad);
smoothedTilt = alpha * rawTilt + (1 - alpha) * smoothedTilt;
float correctedTilt = -(smoothedTilt - calibrationOffset);
arrowImage.setRotation(-correctedTilt);
```

Запуск GPS та отримання швидкості

```
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 1,
this);
float speedKmh = location.getSpeed() * 3.6f;
smoothedSpeed = speedAlpha * speedKmh + (1 - speedAlpha) * smoothedSpeed;
```

Обертання стрілки спідометра

```
float speedAngle = (smoothedSpeed / 180f) * 180f;
speedAngle = Math.max(-90f, Math.min(90f, speedAngle));
speedArrowImage.setRotation(speedAngle);
```

Запит дозволу на використання локаційних даних

```
private void requestLocationPermission() {
    ActivityCompat.requestPermissions(this,
        new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
        LOCATION_PERMISSION_REQUEST_CODE);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions,
                                     @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);

    if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            startLocationUpdates();
        } else {
            Toast.makeText(this, "Потрібен дозвіл на використання GPS",
Toast.LENGTH_LONG).show();
            rawGpsText.setText("GPS: Дозвіл відхилено");
        }
    }
}
```

Скріншоти додатку



Рис. 1. Скріншот 1



Рис. 2. Скріншот 2

Репозиторій

Код було завантажено до репозиторію GitHub. Переглянути його можна за [посиланням](#).

Висновки

Було успішно реалізовано розширення функціональності Android-додатку шляхом інтеграції GPS-даних. Це дозволило додатку не лише показувати поточний нахил автомобіля, але й відображати його швидкість руху у зрозумілому та наочному вигляді.